input is received, processing continues in normal GUI mode. However, if a touch input is received, then processing continues to act **84**.

[**0054**] In act **84**, a location where the touch input was received is generated. The location information can be computed, or it can be explicitly provided by nature of the sensors in the touch input device **58** that monitor for the touch input.

[**0055**] In act **86**, the touch input is correlated to a control field, meaning that the location of the touch input is matched against the location of the nearest control field currently presented on the display device **56**. This act, it is noted, can be performed by the touch-sensitive input overlay module **54** or another module that typically handles a mouse or keyboard entry that moves a cursor into the control field or highlights a dialog window on the GUI. Act **88** can be considered along with act **86**, because in act **86**, a determination is made as to whether a control field exists in the proximity of the touch input. In some cases, no control field will exist and thus no touch-sensitive input overlay will be presented, thus processing will continue to normal mode in act **80**. In others, a default or general purpose touch-sensitive input overlay will be presented on the GUI that assists in general navigation. Nevertheless, receipt of the first touch input typically causes the target control to become the focus for the touch input device **58**.

[**0056**] According to one embodiment, in act **90**, location indicator information is fetched from a data structure (e.g. data structure **66**) stored in memory. The location indicator information assists in determining where on the display device **56** the touch-sensitive input overlay should be presented on the display device **56**. The location indicator information can include placement preferences, as are mentioned above, as well as general rules for preventing partial placement of the touch-sensitive input overlay outside of the visible area on the display device **56**.

[**0057**] In act **92**, the touch-sensitive input overlay is placed on the display device **56** in a location derived from the information from acts **84** and **90**. Next, in act **94**, the system waits for another (or a "second") touch input from the touch input device **58**. Act **94** can be another interrupt driven act, and/or it can be a timing driven act wherein the microprocessor **46** waits for a fixed period of time for a second touch input, and if one is not received then error processing acts **95** occur, such as presentation of a nested touch-sensitive input overlay to prompt a user for a reply or to cancel the touch input (the nested touch-sensitive input overlay being absolutely timed so processing continues regardless of whether a second touch input is received), or simply returning the normal GUI mode.

[**0058**] If a second touch input was received at act **94**, then a test is performed in act **96** to determine whether the second touch input was within the boundary area of the touch-sensitive input overlay. If it was not, then it is ignored or error processing occurs, such as a dialog window prompting the user to re-enter the second touch input because it was out of bounds. However, if the second touch input was within the boundary area of the touch-sensitive input overlay, then the second touch input is correlated to an entry option on the touch-sensitive input overlay in act **98**. Act **98** can include, for instance, correlation of the second touch input to a specific entry option such as depression of a button, key, or navigation guide.

[**0059**] As was the case with acts **86** and **88**, acts **98** and **100** are inter-related. In **100**, if there was not a matching entry option corresponding to the second touch input, then processing continues to an error processing mode substantially similar to the modes described in act **95**—for example, giving a user another opportunity to enter a touch input. However, if a matching entry option is found, then in act **102**, a signal corresponding to the entry option is sent from the touch-sensitive input module **54** to the microprocessor **46** so that the appropriate input operations are entered.

[**0060**] If the entry option involves more than a simple button selection, then this act can take place with the interpreter module **55** within the touch-sensitive input module **54**, or within a similar interpreter module in the application program, or more preferably within the operating system. In such an embodiment, the interpreter module **55** can be invoked on the first touch entry received (so the first entry has two functions: invocation of the interpreter module **55** and selection of a first entry), while subsequent touch input entries into the touch-sensitive input overlay are transformed into corresponding signals matching keyboard or mouse-type entries by the now executing interpreter module **55**.

[**0061**] In act **104**, the subject control field is updated, meaning the transformed signals are committed to the field, thus updating the primary GUI control field with the corresponding input. Next, in act **106**, the touch-sensitive input overlay is closed and in act **108**, processing resumes to normal GUI mode until a next touch input is detected (act **82**).

[**0062**] FIGS. **5**-**7** depict embodiments of touch-sensitive input overlays that can be used over a standard GUI **110** in accordance with the invention. **FIG. 5** depicts an embodiment of a touch-sensitive keyboard input overlay **112**, but a touch-sensitive numeric pad input overlay could also be employed. **FIG. 6** depicts an embodiment of a touch-sensitive navigation input overlay **114**. In each of these embodiments, a parent control delineator **124**, here a cutout from the touch-sensitive input overlay, is shown that is disposed between the touch-sensitive input overlay and the parent control. This parent control delineator, which is shown in each of FIGS. **5**-**7** assists in identifying the parent control that corresponds to the touch-sensitive input overlay.

[**0063**] FIGS. **7**A-B depict a more complicated tree-hierarchy navigation GUI with a touch-sensitive navigation pad input overlay **116**. In this embodiment, the GUI **110** is separated into two adjustable areas **118** and **120**. On the left side, area **118**, a navigation tree is augmented by enlarged touch-input control fields **122**. The fields can be navigated by physical touch directly on the control fields, or by depressing an entry option on the touch-sensitive input overlay **116**. While one side of the screen is active, the touch-sensitive input overlay **116** is placed on the opposite side of the screen. However, once a selection is made, the touch-sensitive input overlay **116** is moved to the other side of the screen. (Note further the placement of the parent control delineator **124**.)

[**0064**] It will be appreciated by one of skill in the art that various functional software or hardware components can be achieved in a single software and/or hardware component or multiple software and/or hardware components. The meth-